

实验十二 设计模块（一）实验报告

实验目的：

深入培养设计原则实践的能力，以人事管理系统为例，评估模块设计并提出更完善的解决方案。

学习依赖注入（dependency injection）技术，并将其应用于人事管理系统，提升代码可测试性、可维护性和灵活性。

掌握协作开发平台的使用，提高团队协作效率。

实验内容：

1. 人事管理系统模块设计评估

评估步骤：

(1)选择评估模块: 确定需要评估的人事管理系统模块：

- 员工管理模块: 负责员工信息的增删改查，包含基本信息、联系方式、工作信息、薪资信息、权限等。
- 薪资管理模块: 负责薪资计算、发放、记录等，包含薪资规则定义、扣款规则定义、薪资单生成、薪资报表统计等功能。
- 考勤管理模块: 负责考勤记录、考勤统计、请假审批等功能。
- 绩效管理模块: 负责绩效目标设定、绩效考核、绩效评价、绩效反馈等功能。

(2)选择设计原则: 选择几个重要的设计原则，并根据模块的功能和复杂程度，重点评估相关原则。：

- 单一职责原则 (SRP): 模块是否只负责一项职责？
评估案例：员工管理模块是否包含权限管理功能？
解决方案：将权限管理功能独立成一个模块，或将员工管理模块拆分为员工信息管理模块和权限管理模块。
- 开闭原则 (OCP): 模块是否对扩展开放，对修改关闭？
评估案例：添加新的员工类型 (例如：实习生) 是否需要修改现有的薪资计算逻辑？
解决方案：使用接口和抽象类，定义通用的薪资计算接口，并为不同员工类型实现不同的薪资计算策略，例如使用策略模式。
- 接口隔离原则 (ISP): 模块是否只依赖于它需要的接口？
评估案例：员工信息管理模块是否依赖于所有员工信息，而实际只用到了部分信息？
解决方案：将员工信息接口细化，例如定义员工基本信息接口、员工联系方式接口、员工工作信息接口等，模块只依赖于所需的接口。
- 依赖倒置原则 (DIP): 模块是否依赖于抽象，而不是具体的实现？
评估案例：薪资计算模块是否直接依赖于数据库操作，而不是抽象的数据库访问接口？
解决方案：使用数据库访问接口，将数据库操作逻辑封装在接口实现类中，薪资计算模块只依赖于数据库访问接口。
- 里氏替换原则 (LSP): 模块是否可以被它的子类替换？
评估案例：临时工的薪资计算逻辑是否可以继承正式员工的逻辑，或者需要单独实现？
解决方案：使用继承或接口实现薪资计算逻辑，并根据不同员工类型的特点实现不同的薪资计算逻辑。

(3)进行评估: 根据选择的原则，对模块进行评估，分析模块是否符合这些原则，以及存在哪些问题。

- 案例分析: 针对每个评估原则，选择一个具体模块进行详细分析，例如针对员工管理模块，分析其是否符合单一职责原则，并提出改进建议。

(4)提出解决方案: 针对评估中发现的问题，提出更完善的改进方案，包括：

- 代码重构: 针对违反 SRP、OCP、ISP 原则的模块进行代码重构，提高代码可读性、可维护性和可扩展性。
- 设计模式应用: 使用策略模式、模板模式、工厂模式等设计模式，提高代码的灵活性和可扩展性。
- 测试用例编写: 针对改进后的模块编写测试用例，保证代码质量和功能正确性。

(5)记录评估结果: 将评估结果记录在文档中，包括模块名称、评估原则、评估结果、解决方案、代码示例、测试用例等。

2. 学习依赖注入

依赖注入应用于人事管理系统：

(1)选择依赖注入框架: 选择适合人事管理系统的依赖注入框架:

- Spring: 功能强大、易于使用、社区活跃, 适合大型项目。
- Guice: 轻量级、高效、性能优秀, 适合小型项目。
- Dagger: 编译时依赖注入, 提高代码效率, 适合需要高性能的项目。

(2)确定依赖关系: 确定人事管理系统中的依赖关系:

- 员工管理模块依赖于数据库访问接口: 员工信息需要存储到数据库。
- 薪资管理模块依赖于员工信息接口: 薪资计算需要用到员工信息。
- 招聘模块依赖于邮件发送接口: 需要发送招聘邮件通知。
- 考勤管理模块依赖于时间服务接口: 需要获取当前时间和日期。
- 绩效管理模块依赖于权限管理接口: 需要判断用户是否有权访问绩效信息。

(3)使用依赖注入框架注入依赖项: 使用依赖注入框架将依赖项注入到模块中:

- 使用构造函数注入: 在模块的构造函数中注入依赖项, 如在员工管理模块的构造函数中注入数据库访问接口。
- 使用属性注入: 在模块的属性中注入依赖项, 如在薪资管理模块的属性中注入员工信息接口。
- 使用方法注入: 在模块的方法中注入依赖项, 如在招聘模块的方法中注入邮件发送接口。

(4)测试依赖注入的效果: 测试依赖注入是否正常工作:

- 模拟依赖项: 使用依赖注入框架模拟依赖项, 例如模拟数据库访问接口、员工信息接口、邮件发送接口, 测试模块的功能是否正常。
- 替换依赖项: 替换依赖项, 例如使用不同的数据库访问接口, 测试模块的功能是否正常。
- 单元测试编写: 编写单元测试用例, 验证依赖注入是否正常工作, 以及模块的功能是否符合预期。